



## UWS Academic Portal

### Remote condition monitoring

Voisin-Grall, Andréa ; Malaolu, Obabiolorunkosi Olaoluwapo; Zhu, Yingbo; Ahmed, Tanveer; Al-Ahmed, Shahriar; Shakir, Muhammad Zeeshan

*Published in:*  
2019 UK/ China Emerging Technologies (UCET)

*DOI:*  
[10.1109/UCET.2019.8881870](https://doi.org/10.1109/UCET.2019.8881870)

Published: 24/10/2019

*Document Version*  
Peer reviewed version

[Link to publication on the UWS Academic Portal](#)

*Citation for published version (APA):*  
Voisin-Grall, A., Malaolu, O. O., Zhu, Y., Ahmed, T., Al-Ahmed, S., & Shakir, M. Z. (2019). Remote condition monitoring: a prototype based on Pycom Development Board FiPy and Pysense. In *2019 UK/ China Emerging Technologies (UCET): UAV, Drone and Vehicle Communication* (2019 ed.). [8881870] IEEE.  
<https://doi.org/10.1109/UCET.2019.8881870>

#### General rights

Copyright and moral rights for the publications made accessible in the UWS Academic Portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

#### Take down policy

If you believe that this document breaches copyright please contact [pure@uws.ac.uk](mailto:pure@uws.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.

# Remote Condition Monitoring: A Prototype Based on Pycom Development Board FiPy and Pysense

Andréa Voisin-Grall, Obabiorunkosi Olaoluwapo Malaolu, Yingbo Zhu, Tanveer Ahmed, Shahriar Abdullah Al-Ahmed, *Student Member, IEEE*, and Muhammad Zeeshan Shakir, *Senior Member, IEEE*

**Abstract**—Internet of Things (IoT) is not new in the market; however, they are becoming more dominant in various operations and applications. These pervasive infrastructures can collect different data in a given environment such as temperature, pressure, light sensitivity, and so on to enable remote condition monitoring. Subsequently, these collected data can be used for many purposes depending on the user's requirement. Currently, there are colossal interests from researchers to know how to use these infrastructures to collect and send data over different protocols to the cloud for efficient remote handling from anywhere in the world. In this paper, we design and implement an infrastructure based on Pycom development board FiPy and sensor shield Pysense to collect and send data to the remote cloud over Wi-Fi and Long Range (LoRa) protocols. The intelligible set up is beneficial for observing and managing data in the cloud.

**Index Terms**—Remote condition monitoring, IoT, Pysense, FiPy, Wi-Fi, LoRa

## I. INTRODUCTION

Internet of Things (IoT) is one of the hot topics nowadays as IoT surrounds our daily life. IoT refers to the interconnection between different objects and devices that can communicate with each other and send data without any human interaction [1]. From a general sensor to a wristwatch, IoT provides solutions to improve human life and industrialization by performing several little tasks smartly. Invented several years ago, IoT is now a vital sector in the industry and become one of the main subjects of research.

IoT sensors can be used to monitor physical quantities remotely with wireless connectivity apparently from anywhere in the world. Wireless technologies are also supporting the scalability of the network. Some highlighted areas are:

- Smart home: IoT sensors are used for smart home applications such as controlling appliances through mobile apps, monitoring utility meters, etc. [2].
- Smart commuting and mobility: Smart commuting, as well as smart mobility, is widely considered the application of IoT sensors for smart cities. The departure and arrival time of the buses are published over the internet these days for mobile apps to suggest commuters when to leave home and plan their journey [3]. Monitoring children movement in a school bus is another excellent

example of IoT application [4]. Early fault detection in railway tracks save lives, reduce accidents, and improve timings of the train schedule [5]. Smart mobility application such as efficient parking facilities, street light control, and traffic management using IoT bring luxury in life with greener impact towards the environment [6].

- Smart city: Smart city based on IoT sensors concept is there for the last few years. The scalability and flexibility of the feature of IoT sensors can monitor many mechanical, electrical, and electronic systems [7]. Another application can be mentioned as smart waste collection from every home. This service monitors the status of the garbage bin and alert the collector truck for scheduling and routing for garbage collection [8].
- e-Health: The health sector is already using IoT widely. Health professionals are now able to monitor the health status of patients from home, hospitals, or their clinic by using e-health system [9].
- Smart farming: Intelligent IoT sensors combined with machine learning are making the agriculture more meticulous by monitoring the crop condition and predicting the suitable time for rain, sowing, and harvesting time. It is an extra privilege for farmers to achieve the highest gain from their investment by reducing the cost of labour and waste of the crop [10].

In this paper, we develop a prototype of IoT sensors to monitor environment remotely by interfacing the sensors via wireless protocols and allowing users to have real-time access to all the data that the sensors are measuring without the constraint of having wires everywhere. We use Pycom development board FiPy and sensor shield Pysense to monitor temperature, humidity, pressure, altitude, and light. Two wireless protocols, name: Long Range (LoRa) and Wi-Fi are used to transmit the sensors data to the cloud via gateways. A dedicated cloud-based dashboard has also been designed to display the data in an appropriate graphical format. The paper is organized as follows. Section II describes state of the art related to this work. Required hardware and software description is given in Section III. We discuss the design and implementation stages in Section IV. Then the results and evaluation are presented in Section V. Finally, we conclude in Section VI.

## II. RELATED WORK

LoRa wireless technology is trending research these days; however, there are only few IoT prototype-based works on

A. V. Grall is with Université d'Aix-Marseille, France. Email: andrea.voisin-grall@etu.univ-amu.fr.

O. O. Malalu, Y. Zhu, T. Ahmed, S. A. Al-Ahmed and M. Z. Shakir are with the School of Computing, Engineering and Physical Sciences, University of the West of Scotland, UK. Email: {B00278805; B00358141; B00325255}@studentmail.uws.ac.uk; {shahriar.al-ahmed; muhammad.shakir}@uws.ac.uk.

Pycom modules using LoRa and Wi-Fi protocols. The authors in [7] have developed a prototype on Pycom modules and presented the principle of operation as well as results of received signal levels on different floors of a building. In [11], the authors have presented a low-powered design of IoT by using Pysense sensor shield, Wasmote, and Raspberry Pi. The authors in [12] used Pycom/Pytrack to collect data and send them to the cloud for big data analysis. Again, the authors in [13] have Pycom LoPy module to collect data from multiple sensors and send data to The Things Network (TTN), which are retrieved by Node-Red to display the data in a dashboard.

### III. HARDWARE AND SOFTWARE REQUIREMENT

In this section, we discuss different tools used to implement our prototype. The selected products are from Pycom Ltd. and their devices and development boards which are compatible with other commonly used hardware Arduino, Raspberry Pi and various protocols such as Wi-Fi, LoRa, Bluetooth Low Energy (BLE), NB-IoT and LTE-M.

#### A. Pysense Sensor Shield

Pysense is sensor shield developed by Pycom to monitor humidity/temperature sensor, barometric pressure sensor, ambient light sensor, a third axis accelerometer [11]. It can be used along with any Pycom development boards such as LoPy (Lora based development board), FiPy (LoRa, Sigfox and dual LTE-M (CAT M1 and NB-IoT) based development board), GPy (cellular LTE CAT M1/NB1 based development board). A typical picture of Pysense sensor shield has been given in Fig.1a. Since this device cannot work alone because it needs other development boards to drive, a FiPy development module has been used for our prototype.

#### B. FiPy

FiPy is a micropython enabled development board with five network protocols: Wi-Fi, LoRa, BLE, Sigfox and dual LTE-M (CAT M1 and NB-IoT) connectivity. It has a powerful processor with a 1km line of sight (LoS) Wi-Fi range and ultra-low-power facility [14]. FiPy can use an appropriate antenna to work properly and provide a high-quality signal. We used the manufacturer recommended antenna for LoRa connectivity to gain long range. Fig.1b displays a FiPy development module from Pycom. FiPy is quite easy to set up as we only need to plug it on with Pysense sensor shield to enable connectivity. Fig.1c shows that Pysense and FiPy are connected.

#### C. Power supply

The development board can be powered up by USB connection or battery for remote installation. We powered up the device by using a USB connection.

#### D. Wireless Router

A generic home broadband wireless router has been used in this experiment to enable Wi-Fi connectivity of the Pysense sensor shield. The wireless SSID and password have been given in the Wi-Fi configuration code in python so that the development board can connect to the Wi-Fi automatically.

#### E. Laird Sentrius RG1xxLoRa gateway

Laird RG1xxLoRa gateway can provide an ultimately scalable, robust, and secure LoRa solutions for end to end management of private Long Range Wide Area Network (LoRaWAN) network. This type of network has been set up for our prototype so that the Pysense sensor shield can transmit the sensor data over LoRa protocols. It is compatible with third-party clouds such as TTN and any LoRaWAN certified client devices.

#### F. PyBytes

Pybytes has been used to see the received data from the Pysense sensor shield for visualizing our data. It is a cloud-based IoT device management platform. The reason for using Pybytes is that it allows connections via Wi-Fi/ Sigfox or LoRa by using LoRa communication methods, i.e., Over The Air Activation (OTAA) and Activation by Personalization (ABP).

#### G. Atom

Atom can be considered as a text editor for building any prototype. It enables the user to communicate with the development board via USB port to install *Pymakr* plugin developed by Pycom for Pycom development boards. This plugin is used to build command line read-eval-print loop (REPL) to sync, upload, download, program, and display output [15].

### IV. DESIGN AND IMPLEMENTATION

The main design of our prototype is to send Pysense sensor shield data through Wi-Fi and LoRa protocols. FiPy has a Wi-Fi protocol with built-in antenna covering 1km (LoS) range [14]. For the LoRa connectivity establishment, the Laird gateway can be connected through Ethernet or Wi-Fi to transport data to the cloud network. We have chosen to connect the Laird gateway through Wi-Fi to increase the wireless range. Fig.2 gives a concept of the architecture of operation using Wi-Fi and LoRa.

#### A. Setting up Wi-Fi Connection

All Pycom devices have WLAN feature built-in. There are many ways to set up Wi-Fi, such as editing the code into the `main.py` file using Atom. The guideline and codes for connecting Pycom device can be found on Pycom website [16]. For our prototype, we set up Wi-Fi connectivity with the aid of Pybytes tool to a Wi-Fi access point (AP) by taking the following instructions in steps:

*Step 1:* Create an account on the `pybytes.pycom.io` website. Click on Add Device and select the device type FiPy. *Step 2:* Select Wi-Fi and make sure it is in the top of the arranged list. There are other options such as Sigfox and LoRa using OTAA or LoRa using ABP.

*Step 3:* In the next step, provide a unique name to the Pysense sensor shield and Wi-Fi credentials (i.e., SSID and Password) for the trusted wireless router that this device will connect to.

*Step 4:* After *Step 3*, the user will get a 32-bit activation token which will be used to link Pysense sensor shield to the

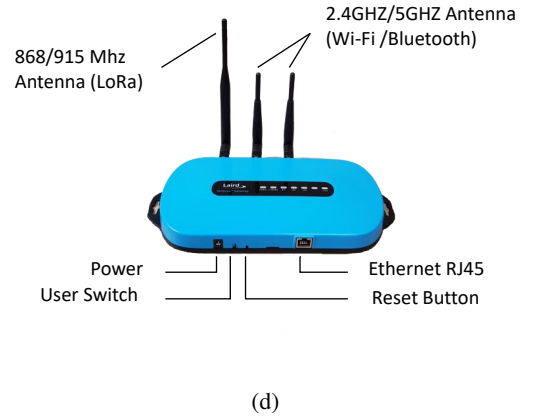
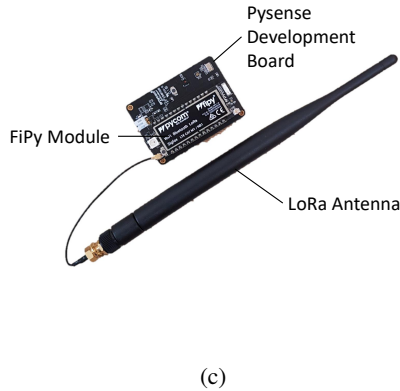
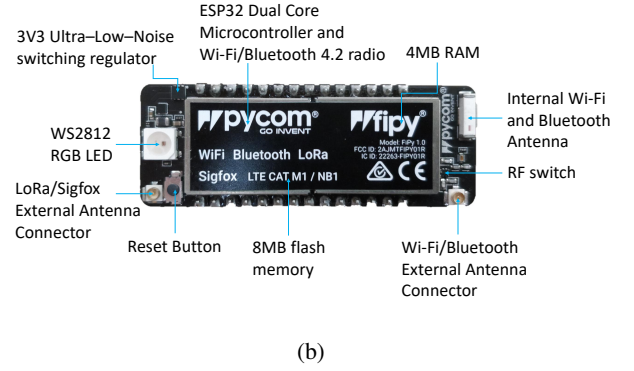
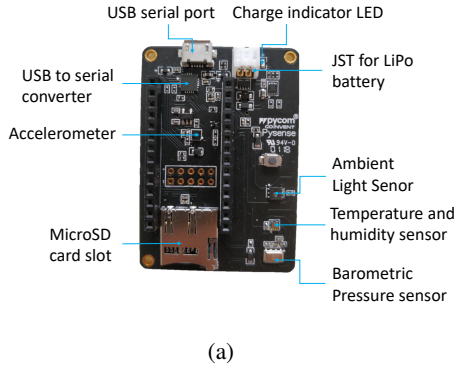


Fig. 1: Hardware used in prototype- 1a Pysense sensor shield , 1b FiPy development module , 1c Fypy and a Pycom plugged together with LoRa antenna, 1d A Sentrius Laird Gateway.

Pybytes dashboard. Download the firmware updater tool and ensure Pysense sensor shield is connected to the computer.

*Step 5:* After downloading the firmware updater tool, it needs to be installed. Open the firmware updater tool after installing and click on the Continue button in the installation interface which will take to the Setup Instruction page, then click Continue again. The next page is the Communication page, where we select the following options:

- High speed transfer
- Erase flash file system
- Force update Pybytes registration

*Step 6:* Then Continue, paste the 32-bit token of *Step 4*, and finally the registration becomes successful. Then click Continue and wait for the firmware to be flashed.

In our prototype, we noticed that `Pybytes_config.json` file installed by the firmware updater tool includes `mqtt.pybytes.pycom.io`. On the other hand, Pybytes runs over an open Message Queuing Telemetry Transport (MQTT) broker [17]. MQTT is a lightweight publish-subscribe messaging protocol for the machine to machine communication. It contains three elements- publisher, broker, and subscriber [18]. Consequently,

Pysense sensor shield is a publisher which is collecting data from the environment and sending them through the Wi-Fi and virtual university network to the broker. Then, the broker forwards the data to the subscriber, which is Pybytes. Based on the understanding of the literature review and `config.json` file, we can say `mqtt.pybytes.pycom.io` is the host MQTT server address for Pybytes.

### B. Setting Up LoRa Connection

To set up the LoRa connection in the Pysense sensor shield, we need first to set up the gateway which is similar to setting up a wireless router. The network uses a star topology as all the end nodes (sensors) connect to the gateway. This gateway captures the LoRa packets which are transmitted by the end nodes (e.g., Pysense sensor shield) and then these packets are scanned and forwarded to their destination. Generally, the nearby gateway captures the transmission of LoRa packets [19]. To set up the hardware Laird Sentrius RG1xx gateway, we need to follow the steps below. Besides, a user guide can be found on the manufacturer website or TTN website [20].

*Step 1:* There are two choices to connect the gateway to the internet, namely, Ethernet cable and Wi-Fi. For the Ethernet connection:

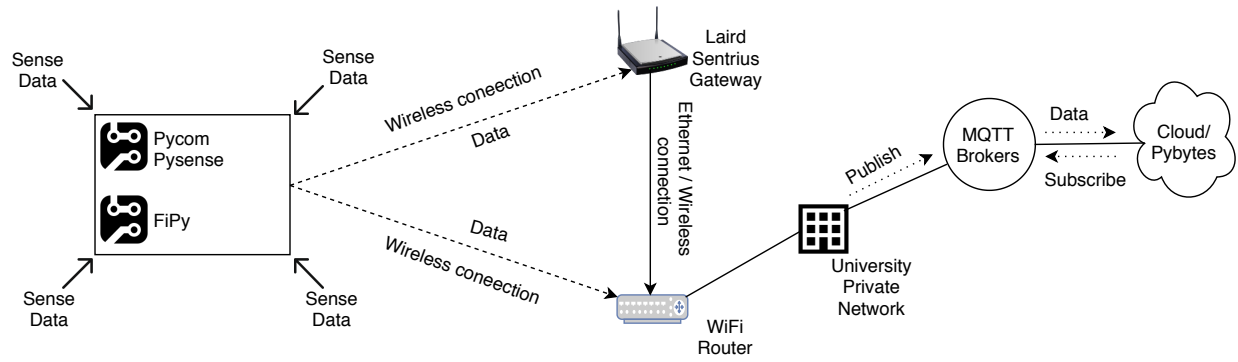


Fig. 2: Architecture of operation using Wi-Fi and LoRa.

- Connect the hardware (i.e., antenna, power cable) and then power up the device.
- Accessing web interface is necessary. Open a web browser and enter `https://rg1xx12345B.local` to access the web interface. 12345B is the last six digit of Ethernet MAC address on the bottom label of Laird gateway
- Log in using default credentials (Username: sentrius Password: RG1xx). These credentials can be changed afterward according to user's requirement.
- Configure the gateway according to the requirement (i.e., changing the LAN settings).
- The Ethernet port is configured to DHCP mode by default. It will connect to the internet immediately when the Ethernet cable is plugged into the router.

For the Wi-Fi connection:

- If the gateway device needs to connect to the internet via a Wi-Fi network, then press the user button for seven seconds, which will change the device mode to AP mode.
- A wireless scan from computer or mobile phone will list the device as `rg1xx12345B` where 12345B is the last 6 digit of Ethernet MAC address.
- Open a web browser and enter `https://192.168..1.1` to access the web interface.
- Log in using default credentials (e.g., Username: sentrius Password: RG1xx).
- Click on the Wi-Fi tab in the main menu of the gateway dashboard and scan for Wi-Fi network, choose a trusted Wi-Fi network and enter the credentials then click connect.

*Step 2:* To set up a device as a LoRa packet forwarder, select LoRa tab from the dashboard of the gateway and then select the Presets tab from the left side menu. Also, select the cloud network from the drop-down menu and select "The Things Network Legacy".

When all the settings are completed, we will see the gateway dashboard that will display all the information about the LAN, Wi-Fi, and LoRa connection.

*Step 3:* Now, register the device which is Laird gateway to a LoRaWAN network: The Things Network (TTN). Sign up to TTN website and click on console and select the gateways. Under the Gateways, select, "Get started by registering one!".

*Step 4:* Extended unique identifier (EUI) of the gateway is required for the registration and can be found at the back of the gateway. In our case, we selected the option "I'm using the legacy packet forwarder" and then we added a description to identify the device. We also selected the frequency for Europe 868 MHz. The gateway will connect to the closest router, which is the `ttn-router-EU/eu.thethings.Network` for lower latency.

*Step 5:* In this step, we are going to use Pybytes cloud, which allows connection of LoRa protocol. To get the prototype working over LoRa, follow the steps of the Wi-Fi connection set up and select the following checkboxes in the firmware updater tool:

- High speed transfer
- Erase flash file system
- Force update LoRa region
- Force update Pybytes registration

After that, we select and rearrange the protocols such as LoRa, Wi-Fi according to the requirements.

The Laird gateway forwards packets that are transmitted by the Pysense sensor shield to the router of the TTN network. The router is connected to one or more brokers and forwards raw data to the nearest broker. Brokers forward uplink data to the correct application as well as forward downlink data to a router. Again, the handler will receive forwarded data from the broker and handle data for one or more applications. It is also responsible for encrypting or decrypting data [21]. Currently, Pybytes works with only TTN network supported MQTT broker [17]. Therefore, the data received in router, brokers, and handler of TTN network platform from the gateway are forwarded to the Pybytes platform.

## V. RESULTS AND EVALUATION

This section presents the results that received from the Pysense sensor shield transmitted by FiPy development board through LoRa gateway and Wi-Fi AP. The data have been recorded in the Pybytes middleware and represented using the widgets which are provided by Pybytes. The main purpose of this prototype is to measure the environmental data using both Wi-Fi and LoRa continually. Pybytes can receive five raw data from five built-in sensors (temperature, humidity, pressure,



Fig. 3: Measured data and graphs from Pybytes: Top row left image shows temperature, right image shows humidity. Bottom row left image shows light intensity, middle image shows air pressure and right image shows altitude.

altitude, and ambient lights), which can be displayed on graphs using five widgets for analysis. The graphs of received data have been presented in Fig. 3.

All the measurements have been recorded outdoor to compare with the standard results and to observe the performance of the Pysense sensor shield. From the temperature graph, it can be seen that the recorded temperature slightly increases over time from 28°C to 30°C approximately as the Pysense sensor shield starts operating. Afterward, the temperature drops but stays in above range. We also compare the recorded results from the built-in temperature sensor in Pysense sensor shield with external DHT22 and DS18B20 temperature sensors under the same conditions. As an example, the temperature has been recorded as 27.6°C and 27.7°C within every minute interval from DHT22 and DS18B20 sensors, respectively. It has also been noticed that the hardware temperature increases over time, which leads to variations in temperature and related parameters.

It can be seen from the humidity graph that the average humidity level has been recorded as approximately 45.5% with fluctuations over time due to changes in temperature readings. It has also been noticed that the humidity level vary with the changes in temperature. It is happening perhaps due to the heat generated from the hardware which drying off the moisture from the surrounding of the sensors. Again, we compared the recorded humidity level from Pysense sensor shield with a DHT22 sensor which measured the humidity level as 48.7%.

Pysense sensor shield has two light sensors which response to two different wavelengths of light and give output in lux [22]. From the light intensity value, it can be seen that the light

intensity has been measured as [11794, 13932] lux, which shows acceptable light intensity under the outdoor testing environment and in comparison with many different lux rating charts found for outdoor daylight environments [23].

The results of barometric pressure show fluctuations in the graph, and the average pressure has been found as 1014.3 Hpa approximately. Generally, standard atmospheric pressure is known as 1013.25 Hpa, which suggests that our recorded result is acceptable for outdoor environments.

Pysense uses current pressure value and sea-level pressure value to calculate altitude. The formula is given below from [24]:

$$h = 44330.77 \left\{ 1 - \left( \frac{p}{p_0} \right)^{0.1902632} \right\} + OFF\_H, \quad (1)$$

where  $p_0 = 101,326$  Pa denotes sea level pressure,  $p$  denotes atmospheric pressure at the testing location,  $OFF\_H$  is altitude data offset which is a register value and  $h$  is altitude in meters.

From the altitude graph, it can be seen that the altitude has been measured as -10 m to -11 m over a range of time interval. The atmospheric pressure is dependent on the weather condition in the testing location. It is suggested to calculate an offset value to achieve accurate results.

Pybytes is a suitable platform to get the five raw values using Wi-Fi or LoRa. However, we were unable to observe other values, for example, RSSI, SNR in dB. These parameters can be seen using The Things Cloud platform with LoRa protocol.

## VI. CONCLUSION

This paper demonstrates how to connect IoT sensors with different wireless technologies. These technologies will create scalability and efficiency to observe information from a remote location. Also, we have discussed the measured results comparing with other sources of actual results. Future work can be mentioned as using technical data to improve the energy efficiency of the IoT sensors since they can be powered up by the battery, extensive study for the different environmental situations, calculating offset values to achieve accurate results for pressure, altitude and temperature. We can also look at integration for data storage and applying machine learning to our data for making predictions. There is a couple of integration Pybytes offers such as Microsoft Azure and Amazon Web Services where IoT analytics and machine learning are offered.

## ACKNOWLEDGEMENT

We thank Dr. Konstantinos Katzis from the School of Science, European University Cyprus, for his valuable guidance and comments that remarkably improved the manuscript.

## REFERENCES

- [1] S. Singh and N. Singh, "Internet of things (IoT): Security challenges, business opportunities & reference architecture for e-commerce," in *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*. IEEE, 2015, pp. 1577–1581.
- [2] T. Malche and P. Maheshwary, "Internet of things (iot) for building smart home system," in *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)*. IEEE, 2017, pp. 65–70.
- [3] S. Sharad, P. B. Sivakumar, and V. A. Narayanan, "The smart bus for a smart city—a real-time implementation," in *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*. IEEE, 2016, pp. 1–6.
- [4] M. Zaki, K. Alhussein, A. Aalquraini, and M. Raad, "Tot school bus: Children safety," 2018.
- [5] O. Jo, Y.-K. Kim, and J. Kim, "Internet of things for smart railway: feasibility and applications," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 482–490, 2017.
- [6] R. Faria, L. Brito, K. Baras, and J. Silva, "Smart mobility: A survey," in *2017 International Conference on Internet of Things for the Global Community (IoTGC)*. IEEE, 2017, pp. 1–8.
- [7] G. Suci, A. L. Petrache, C. Badea, T. Buteau, D. Schlachet, L. Durand, M. Landez, and I. Hussain, "Low-power IoT devices for measuring environmental values," in *2018 IEEE 24th International Symposium for Design and Technology in Electronic Packaging (SIITME)*. IEEE, 2018, pp. 234–238.
- [8] Y. Lykov, M. Bolinova, V. Slobodiuk, A. Lykova, and S. Makovetskyi, "Investigation of potential opportunities for lorawan technology in conditions of urban construction on the example of pycom modules," in *2018 International Scientific-Practical Conference Problems of Informatics. Science and Technology (PIC S&T)*. IEEE, 2018, pp. 543–547.
- [9] S. S. Chaudhari and V. Y. Bhole, "Solid waste collection as a service using iot-solution for smart cities," in *2018 International Conference on Smart City and Emerging Technology (ICSCET)*. IEEE, 2018, pp. 1–5.
- [10] K. Monteiro, É. Rocha, É. Silva, G. L. Santos, W. Santos, and P. T. Endo, "Developing an e-health system based on iot, fog and cloud computing," in *2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion)*. IEEE, 2018, pp. 17–18.
- [11] R. Varghese and S. Sharma, "Affordable smart farming using iot and machine learning," in *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*. IEEE, 2018, pp. 645–650.
- [12] G. Suci, M. Anwar, I. Rogojanu, A. Pasat, and A. Stanoiu, "Big data technology for scientific applications," in *2018 Conference Grid, Cloud & High Performance Computing in Science (ROLCG)*. IEEE, 2018, pp. 1–4.
- [13] O. Debauche, S. Mahmoudi, and S. A. Mahmoudi, "Internet of things: learning and practices. application to smart city," in *2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech)*. IEEE, 2018, pp. 1–7.
- [14] Pycom, "FiPy," 2019, last accessed 08 May 2019. [Online]. Available: <https://pycom.io/product/fipy/>
- [15] Pycom, "REPL," 2019, last accessed 16 June 2019. [Online]. Available: <https://docs.pycom.io/gettingstarted/programming/repl/>
- [16] Pycom, "WLAN," 2019, last accessed 05 May 2019. [Online]. Available: <https://docs.pycom.io/tutorials/all/wlan/>
- [17] Pycom, "Pysense," 2019, last accessed 16 June 2019. [Online]. Available: <https://pycom.io/pybytes-geo-location-beta-release-june/>
- [18] I. Florea, R. Rughinis, L. Ruse, and D. Dragomir, "Survey of standardized protocols for the internet of things," in *2017 21st International Conference on Control Systems and Computer Science (CSCS)*. IEEE, 2017, pp. 190–196.
- [19] J. M. Marais, R. Malekian, and A. M. Abu-Mahfouz, "LoRa and LoRaWAN testbeds: A review," in *2017 IEEE Africon*. IEEE, 2017, pp. 1496–1501.
- [20] The Things Network, "Connecting the rglxx," 2019, last accessed 15 April 2019. [Online]. Available: <https://www.thethingsnetwork.org/docs/gateways/laird/>
- [21] —, "Network architecture," 2019, last accessed 15 June 2019. [Online]. Available: <https://www.thethingsnetwork.org/docs/network/architecture.html#mqtt-handler>
- [22] Pycom, "Pysense," 2019, last accessed 01 June 2019. [Online]. Available: <https://docs.pycom.io/pytrackpysense/apireference/pysense/>
- [23] e-Line Technology, "Lux Light Illumination Chart," 2019, last accessed 28 June 2019. [Online]. Available: <https://elinetechnology.com/support/lux-light-illumination-chart/>
- [24] NXP Semiconductors, "MPL3115A2," 2018, last accessed 22 June 2019. [Online]. Available: <https://www.nxp.com/docs/en/data-sheet/MPL3115A2.pdf>